

THE STATE UNIVERSITY OF ZANZIBAR

(SUZA)


REGNO	BITA/5/21/048/TZ
FIRST NAME	FARHAT ALI OMAR
COURSENAME	WEB
LECTURENAME	MR MASSOUD
TEST	MARKUP TEST

REPORT

❖ The first step spring initializer.

When create spring initializer first must be open the goggle tab or chrome tab to search the spring initializer than to choose the language java ,select maven project, to choose the package and also than to add dependencies.

Meet the Spring team this August at SpringOne.

 **spring** initializr

Project

☐ Gradle - Groovy

☐ Gradle - Kotlin

☒ **Maven**

Language

☒ **Java**

☐ Kotlin

☐ Groovy

Spring Boot

☐ 3.2.0 (SNAPSHOT)

☐ 3.1.3 (SNAPSHOT)

☐ 3.0.10 (SNAPSHOT)

☐ 2.7.15 (SNAPSHOT)

☐ 3.2.0 (M1)

☒ **3.1.2**

☐ 3.0.9

☐ 2.7.14

Dependencies

ADD ... CTRL + B

MySQL Driver **SQL**

MySQL JDBC driver.

—

Spring Web **WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

—

Spring Data JPA **SQL**

Persist data in SQL stores with .Java

—

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

spring.io

Project Metadata

Group	<input type="text" value="com.example"/>
Artifact	<input type="text" value="demo"/>
Name	<input type="text" value="Test"/>
Description	<input type="text" value="Demo project for Spring Boot"/>
Package name	<input type="text" value="com.example.demo"/>
Packaging	<input checked="" type="radio"/> Jar <input type="radio"/> War
Java	<input type="radio"/> 20 <input checked="" type="radio"/> 17 <input type="radio"/> 11 <input type="radio"/> 8

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

Spring Data JDBC **SQL**

Persist data in SQL stores with plain JDBC using Spring Data.

GENERATE CTRL + ↵

EXPLORE CTRL + SPACE

SHARE...

The second step

After create spring boot initializer to generate folder and than to open folder and to create crud MODEL

This is the table to create the get and set in our field that exist.

c > main > java > com > example > demo > model > J modelemployee.java

```
1  package com.model;
2
3  import javax.persistence.Column;
4  import javax.persistence.Entity;
5  import javax.persistence.GeneratedValue;
6  import javax.persistence.GenerationType;
7  import javax.persistence.Id;
8  import javax.persistence.Table;
9
10 @Entity
11 @Table(name="employeeb")
12 public class modelempoloyee {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private long id;
16     @Column(nullable = false)
17     private String fname
18     @Column(nullable = false)
19     private String email;
20
```

```
24     public modelempoloyee(Long id, String fname,String email){
25         this.id = id;
26         this.fname = fname;
27
28         this.email = email;
29
30     }
31
32     public Long getId() {
33         return id;
34     }
35
36     public void setId(Long id) {
37         this.id = id;
38     }
39
40     public String getFname() {
41         return fname;
42     }
43
44     public void setFname(String fname) {
45         this.fname = fname;
46     }
47
48     public String getemail() {
49         return email;
50     }
51
52     public void setemail(String email) {
53         this.email = email;
54     }
55
56 }
```

CONTROLLER

```
c:\main\java\com\controller> controlleremployee.java
1  public class controlleremployee {
2
3  }
4  package com.controlleremployee;
5
6
7  import java.util.List;
8
9  import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.http.ResponseEntity;
11 import org.springframework.web.bind.annotation.CrossOrigin;
12 import org.springframework.web.bind.annotation.DeleteMapping;
13 import org.springframework.web.bind.annotation.GetMapping;
14 import org.springframework.web.bind.annotation.PathVariable;
15 import org.springframework.web.bind.annotation.PostMapping;
16
17 import org.springframework.web.bind.annotation.RequestBody;
18 import org.springframework.web.bind.annotation.RequestMapping;
19 import org.springframework.web.bind.annotation.RestController;
20
21 @CrossOrigin(origins = "**")
22 @RestController
23 @RequestMapping("/employee")
24 public class controller {
25
26     @Autowired
27     private CollegesRepository modelemployeeRepository;
28     @PostMapping("/add")
29     public modelemployee addEmployee(@RequestBody modelempolo
30         return modelemployeeRepository.save(employeebtb);
31     }
```

```

    }
    @GetMapping("/add")
    public List<Colleges> getAllEmployee(){
        return modelempoloyeeRepository.findAll();
    }
    @DeleteMapping("/add/{id}")
    public void delete(@PathVariable long id){
        Repository.deleteById(id);
    }
}

```

SERVICES

```

public class repositoryemployee {
    package com.demo.repository;

    import org.springframework.data.jpa.repository.JpaRepository;

    import com.demo.model.repositoryemployee;

    public interface repositoryemployeeRepository extends Jpa
}

```

After finished the crud go to the API client for me choose postman to test my data.

```
1  }
2  .... "Id": 3,
3  .... "fname": "ali",
4  .... "email": "ali@gmail.com"
5
6  ....
7  }
```

onse



Could not send request